Particle Accelerators
1976, Vol. 7, pp. 213–222

SUPERFISH—A COMPUTER PROGRAM FOR EVALUATION OF RF CAVITIES WITH CYLINDRICAL SYMMETRY

K. HALBACH

Nuclear Science Division, Lawrence Berkeley Laboratory, University of California, Berkeley, California 94720, USA

and

R. F. HOLSINGER

Los Alamos Scientific Laboratory, University of California, Los Alamos, New Mexico 87545, USA

(*Received June 17, 1976*)

The difference equations for axisymmetric fields are formulated in an irregular triangular mesh, and solved with a direct, noniterative method. This allows evaluation of resonance frequencies, fields, and secondary quantities in extreme geometries, and for the fundamental as well as higher modes. Finding and evaluating one mode for a 2000 point problem takes of the order of 10 sec on the CDC 7600.

1 INTRODUCTION

Over the last 10 to 15 years, a number of computer programs have been developed that find the electromagnetic resonance frequency and evaluate the axisymmetric fields in rf cavities with axisymmetric symmetry. The codes that allow this analysis to be made in an essentially arbitrary axisymmetric geometry (see for instance Refs. 1–3) have the following in common: For some geometries, like cavities that have a large diameter compared to their length, and/or for modes higher than the fundamental mode, the convergence rate can be extremely small, or convergence may not be achieved at all. Stated very briefly, the reason for these problems is the fact that in all these codes, an overrelaxation method is used to solve a set of homogeneous linear field equations. The properties of these equations are such that some well developed methods for overrelaxation-factor optimization are not applicable, and it might well be true that the eigenvalues of the matrices for some problems are located in such a way that even an optimized overrelaxation scheme would still result in unacceptably low convergence rates.

To eliminate these problems, we developed the code SUPERFISH that uses a direct, noniterative method to solve a set of inhomogeneous field equations. This code is a combination of some parts of the code RFISH,[4] some new ideas, and the direct solution method used by Iselin in his magnet code FATIMA.[5] In order to give a good overall understanding of SUPERFISH in a limited space, we do not present all detailed formulas, but do include the description of all parts that are conceptually significant, even at the expense of reformulating and/or condensing parts of the cited literature.

In Sections 2 and 3 we discuss separately the structure of the difference equations, and the direct, noniterative method used to solve a set of inhomogeneous linear equations. In Section 4 the basic structure of SUPERFISH is described, and the remaining sections give some details of the theory and of the program as it exists today, and an outline of contemplated future developments.

2 STRUCTURE OF THE DIFFERENCE EQUATIONS IN AN IRREGULAR TRIANGULAR MESH

Inspection of Maxwell's equations shows that for $\partial\mathbf{E}/\partial\phi = 0$, $\partial\mathbf{H}/\partial\phi = 0$, i.e., axisymmetric fields, two independent sets of solutions can exist: one having as only nonzero field components E_ϕ, H_z, H_r; the other, H_ϕ, E_z, E_r. These two solutions are, for equivalent boundary conditions, identical;

we therefore talk only about the latter set. Assuming, without loss of generality, that the magnetic field is proportional to $\cos \omega t$, and the electric field is proportional to $\sin \omega t$, and using suitable units, Maxwell's equations can be written as

$$\text{curl } \mathbf{H} = k\mathbf{E}, \tag{1a}$$

$$\text{curl } \mathbf{E} = k\mathbf{H}, \tag{1b}$$

with $k = \omega/c$, and $\mathbf{H}$ and $\mathbf{E}$ representing the electric and magnetic fields divided by their respective time dependence.

We seek to find numerical solutions for some of the eigenvalues k and associated fields of Eqs. (1a) and (1b) in cylindrical cavities of essentially arbitrary shapes, with $\mathbf{H} = 0$ on the axis and possibly some other parts of the boundary (Dirichlet boundaries), and the electric field perpendicular to the remaining boundaries (Neumann boundaries), implying infinitely conducting walls there.

2.1 *The Mesh*

To solve the differential Eqs. (1a) and (1b), we introduce an irregular triangular mesh[6] in the z-r plane. Figure 1 shows the logical mesh, with mesh points identified by labels K and L, assuming the integer values 1 through $K_{\max} = K_2$, and 1 through L_2. To establish a mesh that can be used to solve the field equations for a particular geometry, defined by its boundaries, the user first assigns boundary coordinates z, r to an arbitrary but reasonable selection of logical points K, L. The mesh generator, described in Ref. 6, then generates a mesh of triangles that is topologically identical to the logical mesh, but has all boundaries defined by mesh lines. The mesh generator finds the z, r coordinates of interior points, for a given set of boundary points, with an iterative process that is similar to a numerical method used to solve Laplace's equation. Figure 2a shows a logical mesh, and Figure 2b a physical mesh, for one half of an Alvarez cavity. In Figure 2a, points on heavily drawn logical lines represent those with assigned z, r coordinates. The two heavily drawn interior lines are used to delineate zones with different mesh point densities. Exterior mesh points, i.e., points inside the drift tube, are not shown since they do not affect the field calculations.

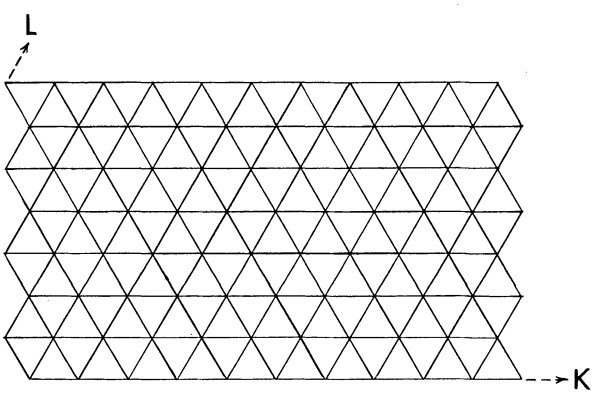

FIGURE 1 Logical triangular mesh.

2.2 *The Difference Equations for Interior Points*

We use the quantity $H = H_\phi$ to describe the rf fields. This somewhat unconventional choice (usually $r \cdot H_\phi$ is used) has the advantage of not requiring any special treatment of the region close to the axis, since H will be proportional to r there, whereas $rH_\phi \sim r^2$ for small r. From Eqs. (1a) and

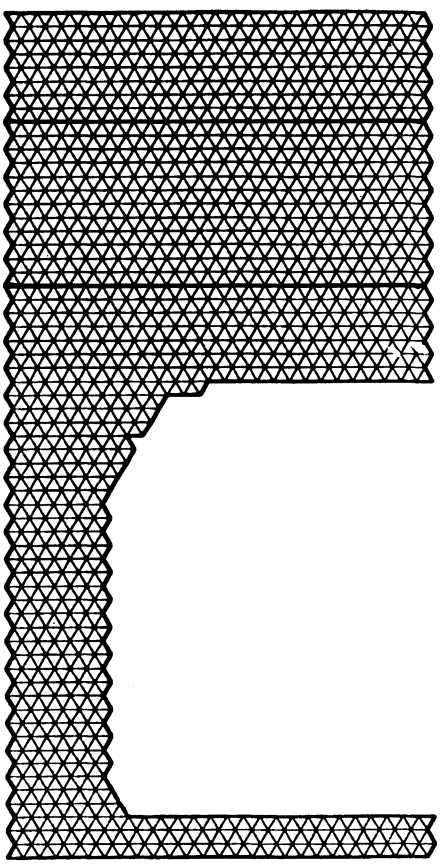

FIGURE 2a Logical mesh for 1/2-Alvarez cavity.

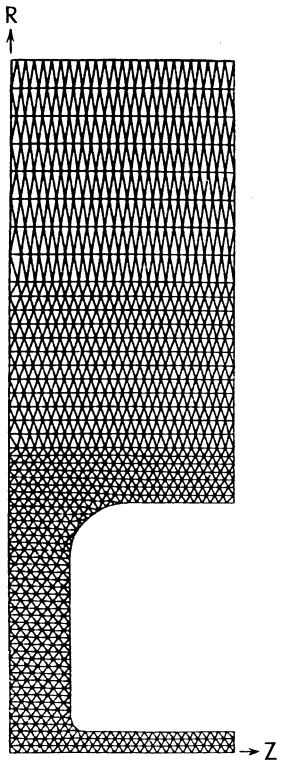

FIGURE 2b Physical mesh for 1/2-Alvarez cavity.

(1b) we obtain as the differential equation for H:

$$\text{curl}(\text{curl } \mathbf{H}) = k^2\mathbf{H}. \tag{2}$$

To derive difference equations for H, we use the procedure described by Winslow[6]: we first introduce a secondary mesh by drawing connecting lines between the "center of mass" of every triangle and the center of each of the three sides of the triangle. As a consequence, every mesh point is now surrounded by a unique twelve-sided polygon. This secondary mesh of dodecagons covers completely the whole problem area, and Figure 3 shows the dodecagon surrounding just one mesh point. The difference equations for H are now obtained by integrating Eq. (2) over the area (in the z-r plane) of one dodecagon at a time. This yields

$$\int \text{curl}(\text{curl } \mathbf{H}) \cdot d\mathbf{a} = \oint \text{curl } \mathbf{H} \cdot d\mathbf{s} = k^2 \int \mathbf{H} \cdot d\mathbf{s}. \tag{3}$$

Assuming that H behaves like a linear function of z and r within every triangle, H inside every triangle is uniquely determined by the values of H at the three corner-mesh points of the triangle.

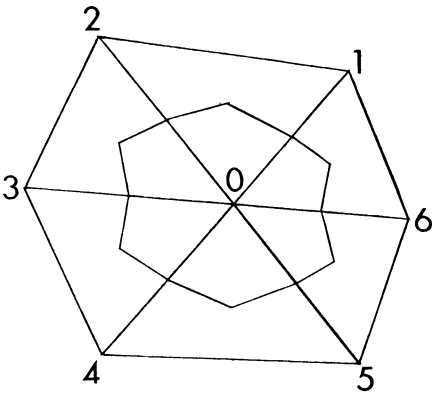

FIGURE 3 Irregular triangular mesh with secondary dodecagon.

The integrals in Eq. (3) can therefore be expressed in terms of the value of H at the "center-mesh point" of the dodecagon and its six nearest logical neighbors, giving a relationship of the following kind

$$\sum_0^6 H_n(V_n + k^2 W_n) = 0, \tag{4}$$

with V_n and W_n depending only on the coordinates z, r of the seven mesh points involved.

Identifying each difference equation with its "center-point," we therefore get one difference equation for H at every interior mesh point.

2.3 *The Treatment of Boundary Points*

Turning now to mesh points on the boundaries of the problem, it is clear that no difference equations are needed for H at boundary points when the boundary conditions require $H \equiv 0$ there. Nevertheless, we have to explore whether or not the difference equations for such points are satisfied. To this end, we consider first Dirichlet-boundary points that are not on the problem axis. This kind of boundary condition can obviously only be imposed as a symmetry condition along a plane defined by $z = \text{const}$. This implies that in the real world a point on one side of this line has an H-value of the same magnitude, but opposite sign, as the symmetrically located point, and the difference equation, Eq. (4), is clearly satisfied for every such boundary point.

This argument cannot be applied without elaboration for points on axis ($r = 0$), and the difference equations resulting from Eq. (3) are in fact not satisfied for those points. To see how this can be interpreted, we can introduce on the

right-hand side of Eq. (1b) a (magnetic) current density term $\mathbf{j}$ that has only an azimuthal component. This gives on the right-hand side of Eq. (3) the additional term kI, where I represents the total current associated with the point under consideration, and assumed to be concentrated there. In our case of axis points, an azimuthal current on the axis is, of course, without consequences, and this whole argument could also be used to lend legitimacy to the application of the symmetry consideration to axis points.

When $\mathbf{E}$ is required to be perpendicular to boundary-mesh lines, we consider that part of the dodecagon surrounding a boundary point that goes through inside problem triangles, i.e., the polygon 0–1′–2′–3′–4′–0 in Figure 4. Since $\mathbf{E} = \text{curl } \mathbf{H}/k$ is required to be perpendicular to lines 4–0 and 0–1,

$$\int_{4'-0-1'} \text{curl } \mathbf{H} \cdot \mathbf{ds} = 0.$$

This means that the difference equation for H at such a boundary point is identical to that for interior points except that only contributions from interior triangles are taken into account.

A very important property of the difference equation, Eq. (4), is the fact that if we use the logical coordinates (indices) K, L to identify mesh points, and if K_0, L_0 are the coordinates of any specific mesh point for which we write down Eq. (4), then the coordinates of the other mesh points contributing to Eq. (4) differ from K_0 and L_0 by not more than ± 1.

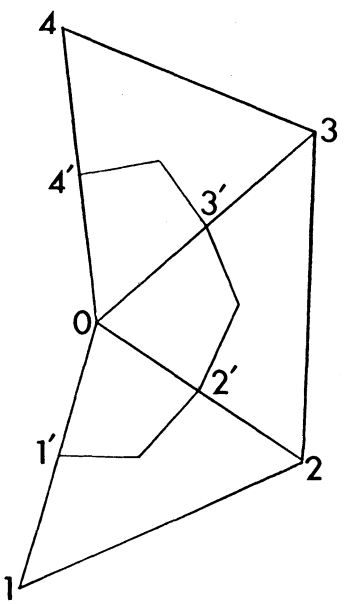

FIGURE 4 Boundary mesh point with neighboring interior mesh points, and secondary polygon.

3 DIRECT, NONITERATIVE SOLUTION OF A SET OF INHOMOGENEOUS LINEAR EQUATIONS

If one wrote down difference Eq. (4) for *all* (i.e., including all boundary and exterior mesh points) H_{KL} of the logical mesh by rows from left to right; and if one also had some inhomogeneous terms, one could write the resulting system of equations in the following form:

$$\begin{pmatrix} a_{11} & a_{12} & & & & & \\ a_{21} & a_{22} & a_{23} & & & & \\ & a_{32} & a_{33} & a_{34} & & & \\ & & & \vdots & & & \\ & & & a_{L_2-1,L_2-2} & a_{L_2-1,L_2-1} & a_{L_2-1,L_2} & \\ & & & & a_{L_2,L_2-1} & a_{L_2,L_2} & \end{pmatrix} \times \begin{pmatrix} \mathscr{H}_1 \\ \mathscr{H}_2 \\ \mathscr{H}_3 \\ \vdots \\ \mathscr{H}_{L_2-1} \\ \mathscr{H}_{L_2} \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ \vdots \\ G_{L_2-1} \\ G_{L_2} \end{pmatrix} \tag{5}$$

In this matrix equation, $\mathscr{H}_1$ represents a column vector with the components H_{K1}, $K = 1 - K_2$; $\mathscr{H}_2$, a vector with components H_{K2}, $K = 1 - K_2$, etc, and the G_n represent the corresponding inhomogeneous terms. The matrix on the left side of Eq. (5) has all zeroes except for the block matrices a_{ij} of size $K_2 \times K_2$. These blocks are also sparse, the diagonal blocks containing only three nonzero elements in every row, and the off-diagonal blocks not having more than three in any row. Deferring to Section 4 the discussion of how we can cast our field evaluation problem in the form of Eq. (5), involving all points of the logical mesh as well as inhomogeneous terms on the right side of Eq. (5), we discuss now the method used to solve Eq. (5).

We first transform all diagonal blocks into unity matrices, and remove all blocks to the left of the diagonal blocks, with the Gaussian block elimination process: we multiply the equations represented by the first row of blocks from the left by a_{11}^{-1}, and

then subtract from the second row the new first row after multiplication from the left by a_{21}. The new set of equations is then the same as the original one, except that $a_{21} = 0$; $a_{11} = \mathrm{I}$; and a_{12}, G_1, a_{22}, and G_2 are now modified. This process is repeated, involving rows 2 and 3, then 3 and 4, etc. The very last step in this process is the multiplication of the last row (modified by the previous step) from the left with the modified block matrix $a^{-1}_{L_2,L_2}$.

Having Eq. (5) rewritten in this form, the last row now represents directly the solution for $\mathscr{H}_{L_2}$. Using this now numerically known vector in row $L_2 - 1$ yields directly the solution for $\mathscr{H}_{L_2-1}$, and continuing this back-substitution process yields the numerical values of all components of all block vectors $\mathscr{H}_n$.

It is important to recognize the fact that this particular fast direct method to solve inhomogeneous linear equations can be used only if they can be cast in the form of Eq. (5), and if the matrix on the left side of Eq. (5) is nonsingular.

4 CALCULATION OF FIELDS AND RESONANCE FREQUENCIES IN SUPERFISH

In order to allow application of the direct linear equation solution described in Section 3, we have to include in an artificial way in the system of equations also those points that are part of the logical mesh, but are external to the actual field solution problem. How this is done, and the treatment of points on Dirichlet boundaries, is discussed in Section 4.1; the creation of the inhomogeneous terms is discussed in Section 4.2, and the resonance frequency determination is discussed in Section 4.3.

4.1 *Treatment of Exterior Points and Points on Dirichlet Boundaries*

The simplest and most practical way to include exterior points without affecting the actual field equations, and without causing the matrix on the left side of Eq. (5) to become singular, is to let the equation for every exterior point read $H_{\text{exterior}} = 0$, and to make all couplings to other equations zero by setting the corresponding coefficients equal to zero also. In other words, if n_0 is the index identifying an exterior point (not a block!) in the overall H-vector on the left side of Eq. (5), one simply sets all elements of row n_0 and column n_0 of the matrix in Eq. (5) equal to zero, with the exception of the n_0, n_0 diagonal element, which is set equal to one. The n_0-element of the inhomogeneous contribution vector on the right side of Eq. (5) is set equal to zero also. The logic of the equation-solving routine is arranged in such a way that the thus-introduced zeroes are actually never used in multiplications, just as the other zeroes in the sparse matrices are never used as multipliers either. Points on Dirichlet boundaries are treated in exactly the same way. However, in contrast to exterior points, their z-r coordinates do enter into the expressions for V_n, W_n in Eq. (4) involving the other point(s) of the triangles that have one or more Dirichlet-boundary points at their corners.

4.2 *Generation of Inhomogeneous Terms for Eq. (5)*

In order to turn the set of homogeneous difference equations [Eq. (4)] into a well-posed set of inhomogeneous field equations, one could be tempted to introduce at one mesh point a driving (magnetic) current, as discussed in Section 2.3. That would be an unwise procedure when one is close to a resonance, since the matrix in Eq. (5) is singular for every resonance frequency, leading, as it must, to infinite fields. Instead, we prescribe that an appropriately chosen off-axis mesh point has the field value one and in effect remove the difference equation for that point from the system of difference equations. To do this without destroying the structure of the field equations, we can proceed in one of the following two ways:

1) If the chosen point is identified by its index n_1 in the overall H vector, we set all matrix elements in row n_1 of the matrix in Eq. (5) equal to zero, except for the diagonal element n_1, n_1, which is set equal to 1. In the vector G on the right-hand side of Eq. (5), all elements are set equal to zero, except the n_1-element is set equal to one. Column n_1 of the matrix is left unchanged.

2) Every matrix element in row n_1 and in column n_1 is set equal to zero, except the n_1, n_1 diagaonal element is set equal to one. The vector on the right-hand side of Eq. (5) is set to equal minus the original column n_1 of the matrix, except for element n_1, which is set equal to one.

The second procedure treats the point with the fixed field value in the same way as exterior points and points on Dirichlet boundaries, and in

addition nonzero terms are generated on the right-hand side of Eq. (5). We therefore use that procedure in the code.

In contrast to the explicit introduction of a driving current, with procedures (1) and (2) the matrix on the left side of Eq. (5) is well conditioned even for resonance frequencies.

If we take the original difference equation for the point with the prescribed field value and solve for the field value at that point, using the solution values of the field at the neighbor points, we will get a value different from the prescribed value, except at resonance. This difference can be interpreted as being proportional to the current I_1 necessary at that point to drive the cavity to the prescribed amplitude at the point with the prescribed field value. For this reason we will refer to this point as the driving point.

4.3 *Resonance Frequency Determination*

The driving current I_1 introduced above depends on k^2 through the coupling coefficients in the difference Eq. (4), and the resonance condition is characterized by

$$I_1(k^2) = 0, \tag{6}$$

since then there is no difference between the value of H_{n_1} as calculated from the difference equation for that point, and the prescribed value used there to solve for the fields; i.e., the difference equations are satisfied for all points of consequence. To find the value(s) of k^2 for which Eq. (6) is satisfied,

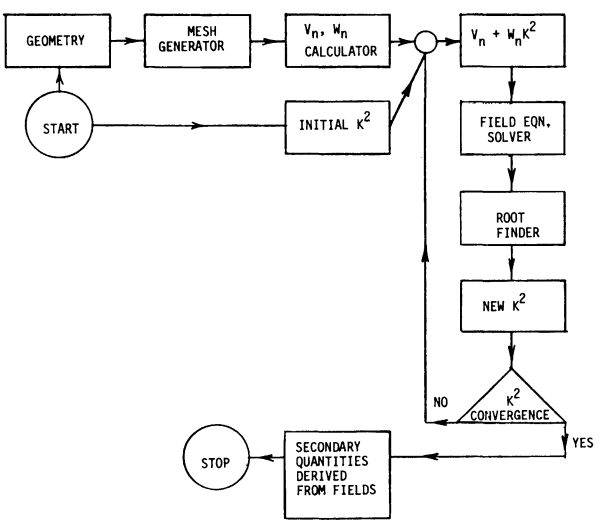

FIGURE 5 Flow diagram of SUPERFISH.

we can combine the above-described "function generator" for $I_1(k^2)$ with a numerical root-finding algorithm, such as the secant method, or a parabola fit method. The latter method is used in the present stage of code development. But we expect that it will be useful to use a root-finding algorithm that takes into account some of the properties of $I_1(k^2)$ that are described in Section 5. Figure 5 depicts a flow diagram of the major parts of SUPERFISH.

5 PROPERTIES OF $I_1(k^2)$ AND INTRODUCTION AND PROPERTIES OF $D(k^2)$

To obtain an understanding of some of the properties of the function $I_1(k^2)$, and later $D(k^2)$, we will go back to the differential equations, Eqs. (1a) and (1b), with Eq. (1b) amended on the right side by the magnetic current density $\mathbf{j}$, assumed to be constant over a small area surrounding the driving point. In the process of deriving some formulas, we have to evaluate integrals like $\int \mathbf{H} \cdot \mathbf{j}_1 \cdot dv$, and set this equal $2\pi r_1 \cdot h_1 \cdot I_1$ where I_1 is the total driving current; r_1, the distance of the driving point from the problem axis; and h_1, the magnetic field averaged over the region where $j_1 \neq 0$. The association between h_1 resulting from this continuum theory and the value of H at a mesh point in the representation by the difference equations is complicated by the fact that h_1 has a logarithmic singularity when the area where $j \neq 0$ is reduced to zero (for fixed I_1). While it seems reasonable to set h_1 equal to H at the driving point, or the value resulting from averaging H over the dodecagon associated with the driving point, it is clear that the quantitative relationships developed below will describe only approximately the relationships between the quantities derived from the difference equations. However, it is also clear that the general behavior of the functions of interest is correctly described by the results derived from the continuum theory below.

Adding the term $\mathbf{j}_1$ to the right side of Eq. (1b) gives

$$\text{curl } \mathbf{E} = k\mathbf{H} + \mathbf{j}_1. \tag{7}$$

Forming the scalar product of both sides of this equation with $\mathbf{H}$, and subtracting from that Eq. (1a), after being multiplied by $\mathbf{E}$, yields:

$$\mathbf{H}\,\text{curl}\,\mathbf{E} - \mathbf{E}\,\text{curl}\,\mathbf{H} \equiv \text{div}(\mathbf{E} \times \mathbf{H}) = \text{k}H^2 + \mathbf{j}_1\mathbf{H} - kE^2.$$

Integrating this over the whole problem volume gives

$$\int \mathrm{div}(\mathbf{E} \times \mathbf{H})\mathrm{d}v \equiv \int (\mathbf{E} \times \mathbf{H}) \cdot \mathrm{d}\mathbf{a} = 2\pi r_1 h_1 I_1 - k \int (E^2 - H^2)\mathrm{d}v. \tag{8}$$

Since $\mathbf{E} \times \mathbf{H}$ is either zero on the problem boundary, or perpendicular to the boundary normal, $\int (\mathbf{E} \times \mathbf{H})\mathrm{d}\mathbf{a} = 0$, and we get

$$D(k^2) \equiv \frac{2\pi r_1 h_1 k I_1}{\int H^2 \, \mathrm{d}v} = R(k^2) - k^2, \tag{9}$$

$$R(k^2) = \frac{\int k^2 E^2 \mathrm{d}v}{\int H^2 \, \mathrm{d}v} = \frac{\int (\mathrm{curl}\, \mathbf{H})^2 \, \mathrm{d}v}{\int H^2 \, \mathrm{d}v}. \tag{10}$$

The new function $D(k^2)$ has the property that its value does not depend on the scaling of h_1, or I_1 if that is the quantity that one wants to consider as the primary variable.

To obtain more information about the behavior of $I_1(k^2)$, we now calculate $\mathrm{d}I_1/\mathrm{d}(k^2) = I_1'$. To this end, we take the derivatives with respect to k^2 of Eqs. (1a) and (7). Indicating derivatives with respect to k^2 by primes, we get

$$\mathrm{curl}\, \mathbf{H}' = k\mathbf{E}' + \mathbf{E}/2k \tag{11}$$

$$\mathrm{curl}\, \mathbf{E}' = k\mathbf{H}' + \mathbf{H}/2k + \mathbf{j}_1'. \tag{12}$$

It should be noted that for our procedure of field evaluation, $\mathbf{H}' = 0$ on Dirichlet boundaries, because $\mathbf{H} = 0$ there for all k^2. Similarly $\mathbf{E}'$ is perpendicular to Neumann boundaries since the component of $\mathbf{E}$ parallel to a Neumann boundary is zero for all k^2. We now consider

$$\mathrm{div}(\mathbf{E} \times \mathbf{H}' - \mathbf{E}' \times \mathbf{H}) \equiv \mathbf{H}' \cdot \mathrm{curl}\, \mathbf{E} - \mathbf{E} \cdot \mathrm{curl}\, \mathbf{H}' - \mathbf{H} \cdot \mathrm{curl}\, \mathbf{E}' + \mathbf{E}' \, \mathrm{curl}\, \mathbf{H}.$$

Using for the curl expressions the appropriate right sides of Eqs. (1a), (7), (11) and (12) yields

$$\mathrm{div}(\mathbf{E} \times \mathbf{H}' - \mathbf{E}' \times \mathbf{H}) = \mathbf{H}' \cdot \mathbf{j}_1 - \mathbf{H} \cdot \mathbf{j}_1' - (E^2 + H^2)/2k.$$

Integrating this over the problem volume gives, as in Eq. (8), zero on the left side, yielding

$$2\pi r_1 k(h_1' I_1 - h_1 I_1') = \int (E^2 + H^2)\mathrm{d}v/2. \tag{13}$$

We intentionally made no *a priori* assumptions whether we consider h_1 or I_1 fixed when k^2 is changed. However, for the case considered so far, $h_1' = 0$, and we can immediately deduce the following conclusions from Eq. (13):

$$h_1 I_1' < 0 \quad \text{(Foster's theorem)}. \tag{14}$$

This means that for fixed h_1, between every two resonances ($I_1(k^2) = 0$) $I_1(k^2)$ must have a singularity such that the sign of $I_1(k^2)$, and therefore also of $D(k^2)$, changes.

At a resonance, $\int E^2 \, \mathrm{d}v = \int H^2 \, \mathrm{d}v$ [see Eqs. (9) and (10)], giving $\int H^2 \, \mathrm{d}v$ on the right side of Eq. (13). We therefore get from Eqs. (13), (9), and (10) at a resonance ($I_1 = 0$):

$$\frac{2\pi r_1 h_1 k I_1'}{\int H^2 \, \mathrm{d}v} = D'(k^2) = R'(k^2) - 1 = -1. \tag{15}$$

Since $I_1(k^2)$ has a singularity between resonances, it is more convenient to study $D(k^2)$ in the vicinity of these singularities. To this end, we first consider $R(k^2)$. According to Eq. (9), $R(k^2) = k^2$ at every resonance, and $R' = 0$ at resonance follows from Eq. (15). Since R cannot be negative, $R(k^2)$ must look qualitatively as indicated in Figure 6 and $R - k^2 = D(k^2)$ as shown in Figure 7. An important consequence is that between resonances, $D(k^2)$ goes through zero, and this sign change must take place where $I_1(k^2)$ has a singularity.

To study $D(k^2)$ in the vicinity of this root of $D(k^2)$ that does *not* represent a resonance, we take advantage of the fact that $D(k^2)$ is independent of the scaling of the field and current quantities. We can therefore consider I_1 as given and kept constant, and consider h_1 as the k^2-dependent

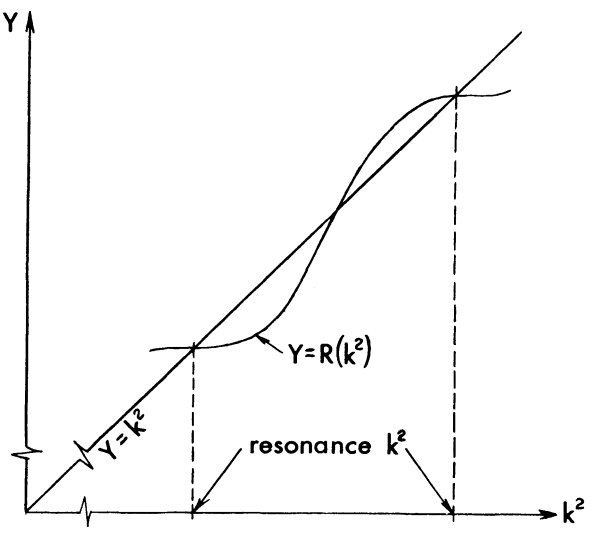

FIGURE 6 Graphical representation of properties of $R(k^2)$.

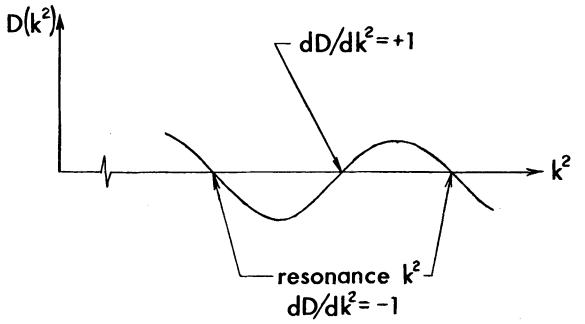

FIGURE 7 Graphical representation of properties of $D(k^2)$

quantity that causes $D(k^2) = 0$. At this "between-resonance root," it follows from Eqs. (9) and (10) that $\int E^2 \, dv = \int H^2 \, dv$, giving again $\int H^2 \, dv$ as the right side of Eq. (13). We therefore get from that equation

$$\frac{2\pi r_1 k I_1 h'_1}{\int H^2 \, dv} = D'(k^2) = R'(k^2) - 1 = 1. \quad (16)$$

A possible use of Eqs. (15) and (16) will be briefly described at the end of Section 6.3.

6 PROGRAM STATUS AND FUTURE DEVELOPMENT

The computer program was originally written for the CDC 7600 operating under the Livermore Time-Sharing System (LTSS), and we describe here that particular version.

6.1 *Computer Time and Storage Requirements*

The CPU time required for a field evaluation is dominated by the time required to invert the block matrices. For the system of equations described at the beginning of Section 3, the time used for inversion of the block matrices is proportional to $K_2^3 \cdot L_2$. When $K_2 > L_2$, the difference equations are arranged along columns of the logical mesh, leading to this expression for the CPU time

$$T = T_1 N^2 \varepsilon, \quad (17)$$

with N representing the total number of logical mesh points, and ε the smaller of the two numbers K_2/L_2, L_2/K_2. For the CDC 7600 under LTSS, $T_1 \approx 0.75$ μsec.

With the present system to find the roots of $I_1(k^2)$, it takes 3 to 6 field iterations to determine a resonance frequency accurately.

The storage requirements for the program are approximately $11 \cdot N$ exclusive of the memory required for the modified off-diagonal block matrices, needed for the back-substitution. These matrices represent $N^{3/2} \cdot \varepsilon^{1/2}$ words, too much to be accommodated in core for $N > 1500$. For larger problems, the disk has to be used. However, S. B. Magyary (LBL) has pointed out that one needs to store only two such matrices when one has to calculate only $I_1(k^2)$ (and not the complete field map), provided the driving point is associated with the last row of block matrices on the left side of Eq. (5). In that case, the large amount of storage is not needed until one has a converged resonance frequency.

6.2 *Accuracy*

Since we know from our experience with the RFISH code and the magnet code POISSON that the program is unlikely to have problems related to curved boundaries, we have made analytically testable runs so far only for empty pill-box cavities.

To see whether this code has any problems with extreme geometries, we ran an empty box of 5 cm length and 150 cm radius with 1267 points. Without any difficulty, the code returned the fundamental frequency correct to all five printed digits.

Much more extensive runs were made for an empty box 60 cm long and a radius of 88 cm. The mesh point separation was 2 cm in both the axial and radial direction, giving a total of 1395 points. The fundamental frequency of this cavity is 130.389 MHz and is reproduced by the code with an error of 1 part in 10,000, while the stored energy is reproduced to an accuracy of 1 part in 3000. A much more severe test is the evaluation of higher modes. Resonance frequency number eight is 582.44 MHz, and is returned by the code as 583.59 MHz; the stored energy calculated by the code is 3% smaller than the correct value. Figure 8 shows the pattern of electrical field lines (rH = const) for this mode. It should be noted that the distance between an extreme value of rH and the next axial node is only 7.5 mesh spacings. Modes 29 and 30 represent an even more extreme test The analytical frequencies are 1179.9 MHz and 1186.3 MHz, and the code-produced frequencies are 1183.0 MHz and 1196.6 MHz, while the energy of these modes is off by approximately 10% Considering the fact that mode 29 has six radia and one axial nodes, and mode 30 has three radia

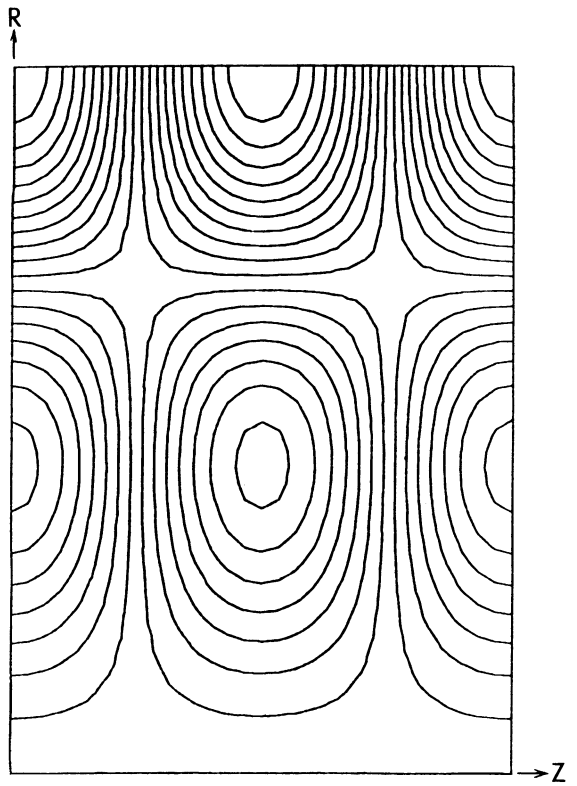

FIGURE 8 Electric field lines (rH = const) for mode No. 8 in test cavity.

and four axial nodes, these numbers are surprisingly good. The closeness of the two resonances did not cause any problems. "Turning on" the partial and complete pivoting of the matrix inversion routines, or iterating on the field residuals of the solution of the difference equations, did not change any of these numbers. However, increasing the number of mesh points caused a marked improvement of the accuracy of the frequency and the stored energy, indicating that the numerical errors are due to mesh size, and not round-off errors.

6.3 *Secondary Quantities, and Near Future Developments*

The program calculates now, or will calculate in the very near future, the following secondary quantities: stored energy; transit time factors; energy dissipated on designated surfaces; $|H|_{\max}$, $|E|_{\max}$ on designated surfaces; shunt impedance; Q; and frequency perturbation by drift tube stems.

We also plan to calculate and print out coefficients that indicate how the movement of designated surfaces perturbs the resonance frequency. These quantities were calculated by RFISH and proved extremely valuable.

To simplify the work on high-order modes, we intend to generate printout plots of node lines (i.e., $H = 0$-lines) and/or plots of points with local extrema of rH, and $I_1(k^2)$ and $D(k^2)$ plots.

To simplify the design of cavities that have to have a predetermined resonance frequency, we intend to run the code with that fixed frequency (possibly modified by drift tube stems) and to accomplish $I_1 = 0$ by moving or deforming a designated boundary. The techniques necessary to do this are already used in the magnet design code MIRT[7] and can easily be incorporated in SUPERFISH.

To reduce the number of iterations necessary to find a resonance frequency, we plan to employ a root-finding routine that uses the properties of $D(k^2)$ expressed by Eqs. (15) and (16). If this code is used extensively to find high-order modes, it might also be profitable to attempt to develop a mode pattern analysis and prediction routine.†

6.4 *Advantages of SUPERFISH*

The main advantage of the code is the capability to solve problems that other codes cannot solve at all, or only with great expenditure of computer time. In addition, the code is quite fast, requiring only about 1 sec per iteration on the frequency for the test problem discussed above. With five iterations and the time used to calculate miscellaneous other quantities, one has a complete solution in 6 sec. The irregular triangular mesh, while not allowing as many mesh points as a square mesh, has the advantage of allowing the definition of boundaries by mesh lines, and to produce a mesh with a large density of mesh points in regions where the problem requires high resolution.

6.5 *Disadvantages of SUPERFISH*

The drawback associated with the irregular triangular mesh is the fact that one has to generate such a mesh. This extra step can slow down the

† Since submission of this report for printing, the more sophisticated root-finding routine has been developed and is working very well. Work on the mode prediction algorithm has started and looks very promising.

total process of receiving the desired answers. This problem has been partly reduced by the creation of the code AUTOMESH, developed by one of us (R.F.H.) while at CERN. This code optimizes automatically the coordination between space-boundary coordinates and logical coordinates, provided that one is satisfied with a uniform mesh point density in a limited number of distinct regions. At the time of writing this paper, an effort is being undertaken at LASL by D. Swenson, W. Jule, and one of us (R.F.H.) to improve the whole process of data input and mesh generation.

There is one basic drawback associated with the necessity of having a driving point in the problem: if one happend to choose its location such that it is on a $H = 0$ line for the problem under consideration computational problems would result. For that reason, it is advisable to put the driving point on a Dirichlet boundary. When the code detects the computational difficulty, it can switch the driving point to a more favorable neighboring point on the boundary, thus eliminating the problem.

ACKNOWLEDGEMENTS

We thank Dr. T. Elioff, Mr. A. Faltens, Dr. L. J. Laslett, Mr. S. B. Magyary at LBL, and Dr. R. A. Jameson, Ms. S. Johnson, Dr. W. E. Jule; Dr. E. A. Knapp, and Dr. D. A. Swenson at LASL, for discussions and/or support of this work. This work was done with support from the U.S. Energy Research and Development Administration. Any conclusions or opinions expressed in this report represent solely those of the authors and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the U.S. Energy Research and Development Administration.

REFERENCES

1. T. Edwards, MURA Report 622 (1961), unpublished (MESSYMESH).
2. H. C. Hoyt, *Rev. Sci. Instrum*, **37**, 755 (1966) (LALA).
3. M. Bell, G. Dôme, *Proc. 1970 Proton Lin. Acc. Conf.*, p. 329.
4. Developed in 1969–70 by K. Halbach, R. F. Holsinger, R. B. Yourd specifically for analysis of cylindrical rf cavities with large diameter-to-length ratio.
5. C. Iselin, to be published.
6. A. Winslow, *J. Comp. Phys.* **1**, 149 (1966).
7. K. Halbach, *Proc. of 2nd Intern. Conf. on Magnet Technology*, Oxford 1967, p. 47.